

CROSS-REFERENCE TO RELATED APPLICATION(S)

15

The present invention relates generally to telecommunications systems, and more particularly, to a system for interfacing telephony devices with DOCSIS compatible networks.

Traditional dial-up modems provide online access through the public telephone network at up to 56 Kbps (equal to 56,000 bits per second). A cable modem, on the other hand, provides users with high-speed Internet access through a cable television network. Cable modem is capable of providing data rates as high as 56 Mbps, and is thus suitable for high speed Internet access, digital television (such as pay-per-view) and digital telephony.

In one aspect of the present invention, a method of synchronizing data clocked by a first clock to a second clock includes generating a clock error signal as a function of one or

09343460

1 systems with a cable modem termination system CMTS), defined as
a line card, disposed at the headend;

FIG. 3 is a is a system block diagram of an exemplary
network gateway in accordance with a preferred embodiment of the
5 present invention;

FIG. 4 is a graphical depiction of the chaining mode of
operation of the system direct memory access controller in
accordance with a preferred embodiment of the present invention;

FIG. 5 is a system block diagram of a DOCSIS downstream
10 demodulator in accordance with a preferred embodiment of the
present invention;

FIG. 6 is a system block diagram of a DOCSIC upstream
modulator in accordance with a preferred embodiment of the
present invention;

15 FIG. 7 is a system block diagram of a DOCSIS media access
controller (MAC) in accordance with a preferred embodiment of the
present invention;

FIG. 8 is a system block diagram of an Ethernet transceiver
for interfacing the network gateway with Ethernet devices in
20 accordance with a preferred embodiment of the present invention;

FIG. 9 is a system block diagram of an Ethernet media access
controller (MAC) in accordance with a preferred embodiment of the
present invention;

FIG. 10 is a system block a Universal Serial Bus (USB)
25 controller in accordance with a preferred embodiment of the
present invention;

FIG. 10A is a system block a MAC for a USB controller;

FIG. 11 is a block diagram of the analog front end for
interfacing the analog processor with an external subscriber line
30 interface circuit (SLIC) in accordance with a preferred
embodiment of the present invention;

FIG. 11A is a block diagram of an external interface
between the analog front end and the subscriber line interface
circuit (SLIC) in accordance with a preferred embodiment of the
35 present invention;

1 FIG. 12 is a block diagram of the audio processor that
interfaces the voice and data processor with external subscriber
line circuits (SLICs) via the analog front end in accordance with
a preferred embodiment of the present invention;

5 FIG. 13 is a block diagram of a ring generator that
synthesizes a reference waveform that is utilized by external
SLICs to ring telephony devices in accordance with a preferred
embodiment of the present invention;

10 FIG. 14 is a system block diagram of a network gateway for
interfacing between a hybrid fiber coaxial (HFC) network and a
switched circuit network and a packet based network in accordance
with a preferred embodiment of the present invention;

15 FIG. 14A is a block diagram of a timing recovery system for
synchronizing the timing regeneration circuit clock of the
network gateway to the CMTS clock in accordance with a preferred
embodiment of the present invention;

20 FIG. 15 is a block diagram of a network gateway including
a voice synchronizer for synchronizing voice data signals between
telephony devices on the near and far ends of a HFC network in
accordance with a preferred embodiment of the present invention;

FIG. 16 is a graphical depiction of fractional interpolation
and decimation of a digitized analog voice signal in accordance
with a preferred embodiment of the present invention;

25 FIG. 17 is a general block diagram of a voice synchronizer
that generates an error signal which is used to polyphase re-
sample the input voice signal so as to synchronize the near end
signal with the far end signal in accordance with a preferred
embodiment of the present invention;

30 FIG. 17A is a block diagram of clock divider circuits for
generating various clocks for use within the network gateway in
accordance with a preferred embodiment of the present invention;

FIG. 17B is a timing diagram illustrates the offset in input
and output sample counts in an buffer overflow condition;

35 FIG. 17C graphically illustrates the zero buffer padding
required between data points in the data buffer and the low pass

presence of far end speech to control the adaptation of a
adaptive filter in accordance with a preferred embodiment of the
present invention;

FIG. 20 is a block diagram of an echo canceller which avoids delays in the detection of near end speech that are typically associated with conventional echo cancellers and utilizes the delay associated with a decimator and high pass filter to provide a look ahead capability so that filter adaptation may be disabled in advance of the arrival of near end speech in accordance with a preferred embodiment of the present invention;

FIG. 20A is a block diagram of an echo canceller that utilizes a delay buffer to provide a look ahead capability so that filter adaptation may be disabled in advance of the arrival of near end speech in accordance with a preferred embodiment of the present invention;

FIG. 21 is a block diagram of method for reducing the amplitude level of near end speech prior to double talk detection in accordance with a preferred embodiment of the present invention;

FIG. 22 is a block diagram of a signal processing system implemented with a programmable digital signal processor (DSP) software architecture for interfacing telephony devices with packet based networks in accordance with a preferred embodiment of the present invention;

FIG. 23 is a block diagram of the MTA architecture for interfacing the voice processor DSP with the MIPS core processor of the network gateway in accordance with a preferred embodiment of the present invention;

FIG. 24 is a block diagram of the architecture for gateway control protocol logic which interfaces with an external control element, called a call agent or call management server (CMS), to terminate and generate call signaling from and to the voice and data processor in compliance with the network-based call signaling (NCS) protocol specification, in accordance with a preferred embodiment of the present invention;

FIG. 25 is a block diagram of method for converting between real time protocol packets required by the PacketCable exchange specification and the xChange packet format utilized by the voice and data processor and vice versa in accordance with a preferred embodiment of the present invention;

FIG. 26 is a block diagram of software messaging interface between the host DSP and the voice and data processing software in accordance with a preferred embodiment of the present invention;

FIG. 27 is a block diagram of channel associated signaling service logic for exchanging commands and events between the host MTA call client and standard commercial analog loop/ground start devices such as for example plain old telephone sets in accordance with a preferred embodiment of the present invention;

FIG. 28 is a block diagram of the software architecture operating on the DSP platform of FIG. 22 in accordance with a preferred embodiment of the present invention;

FIG. 29 is state machine diagram of the operational modes of a virtual device driver for packet based network applications in accordance with a preferred embodiment of the present invention;

FIG. 30 is a system block diagram of a signal processing system operating in a voice mode in accordance with a preferred embodiment of the present invention;

FIG. 31 is a system block diagram of a signal processing system operating in a real time fax relay mode in accordance with a preferred embodiment of the present invention; and

FIG. 32 is a system block diagram of a signal processing system operating in a modem relay mode in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION

In an exemplary embodiment of a cable modem system, a headend or cable modem termination system (CMTS) is located at a cable company facility and functions as a modem which services

maintain timing synchronization between the cable modem transmitter and the CMTS receiver, for proper reception of the communicated information. During continuous communications, timing recovery is a more straightforward process since signal acquisition generally only occurs at the initiation of such communications. Thus, acquisition is generally only performed in continuous receivers once per continuous transmission and each continuous transmission may be very long.

However, the burst communications inherent to TDMA systems entail periodic and frequent reacquisition of the signal. That is, during TDMA communications, the signal is reacquired for each separate burst transmission being received.

Referring now to FIG. 1, a hybrid fiber coaxial (HFC) network 10 facilitates the transmission of data between a headend 12, which includes at least one cable modem termination system, and a number of homes 14, each of which contains a cable modem. Such hybrid fiber coaxial networks are commonly utilized by cable providers to provide Internet access, cable television, pay-per-view and the like to subscribers.

Approximately 500 homes 14 are in electrical communication with each node 16, 34 of the hybrid fiber coaxial network 10, typically via coaxial cables 29, 30, 31. Amplifiers 15 facilitate the electrical connection of the more distant homes 14 to the nodes 16, 34 by boosting the electrical signals so as to desirably enhance the signal-to-noise ratio of such communications and by then transmitting the electrical signals over coaxial cables 30, 31. Coaxial cable 29 electrically interconnects the homes 14 with the coaxial cables 30, 31, which extend between amplifiers 15 and nodes 16, 34. Each node 16, 34 is electrically connected to a hub 22, 24, typically via an optical fiber 28, 32. The hubs 22, 24 are in communication with the headend 12, via optical fibers 20, 26. Each hub is typically capable of facilitating communication with approximately 20,000 homes 14.

The optical fibers 20, 26 extending intermediate the headend 12 and each hub 22, 24 defines a fiber ring which is typically capable of facilitating communication between approximately 100,000 homes 14 and the headend 12. The headend 12 may include video servers, satellite receivers, video modulators, telephone switches and/or Internet routers 18, as well as the cable modem termination system. The headend 12 communicates via transmission line 13, which may be a T1 or T2 line, with the Internet, other headends and/or any other desired device(s) or network.

In an exemplary embodiment of the present invention, a network gateway may facilitate on line and off line bi-directional communication between a number of near end data or telephony devices and far end data termination devices via a cable modem termination system. An exemplary topology is shown in FIG. 2 with a network gateway 11a providing an interface to HFC network 10a for a telephone 53a, fax machine 54 and modem 53c through a subscriber loop interface circuit (SLIC) 50. The network gateway 11a also provides an interface to the 10a HFC network for a telephone 53b, a fax machine 54b and a modem 55b through our audio processor 52. A local area network (LAN) 46 and a universal synchronous bus (USB) 44 can also be provided access to the HFC network 10a via the network gateway 11a. The HFC network is coupled to a CMTS line card 42 in the CMTS 12. The CMTS card 12 is coupled a packet based network router 40 to determine whether the communication will be transported via a far end HFC network 10b, a far end PSTN network 57 or the Internet 43. In the case of a far end PSTN network a PSTN gateway 58 provides an interface between a far end PSTN device 56 and a PSTN network 57 connected to the CMTS 12.

In the case of a far end HFC network, a network gateway 11b provides an interface between the far end data termination devices 56B and the far end HFC network 10b connected to the CMTS 12. As those skilled in the art will appreciate, the far end data termination devices 56 can include a variety of telephony

transparent bi-directional IP traffic between devices operating on a USB such as for example a PC workstation, server printer or other similar devices and the far end data terminating device. Additionally, an I.E.E 802.3 compliant media independent interface (MII) 110 in conjunction with an Ethernet MAC 134 also provide bi-directional data exchange between devices such as, for example a number of PCs and or Ethernet phones and the far end data terminating device. A voice and data processor 160 is used for processing and exchanging voice, as well as fax and modem data between packet based networks and telephony devices.

The QAM downstream demodulator 100 may utilize either 64 QAM or 256 QAM in the 54 to 860 MHz bandwidth to interface with the CMTS. The QAM downstream demodulator 100 accepts an analog signal centered at the standard television IF frequencies, amplifies and digitizes the signal with an integrated programable gain amplifier and A/D converter. The digitized signal is demodulated with recovered clock and carrier timing. Matched filters and then adaptive filters remove multi-path propagation effects and narrowband co-channel interference. Soft decisions are then passed off to an ITU-T J.83 Annex A/B/C compatible decoder. The integrated decoder performs error correction and forwards the processed received data, in either parallel or serial MPEG-2 format to a DOCSIS Media Access Controller (MAC) 112.

The output of the downstream demodulator100 is coupled to the DOCSIS MAC 112. The DOCSIS MAC 112 may include baseline privacy encryption and decryption as well as robust frame acquisition and multiplexing with MPEG2-TS compliant video and audio streams. The DOCSIS MAC 112 implements the downstream portions of the DOCSIS protocol. The DOCSIS MAC 112 extracts DOCSIS MAC frames from MPEG-2 frames, processes MAC headers, and filters and processes messages and data.

Downstream data packets and message packets may be then placed in system memory 114 by a SDRAM interface 116 via the internal system bus 118. The SDRAM interface 116 preferably

1 interfaces to a number of off the shelf SDRAMs which are provided
to support the high bandwidth requirements of the Ethernet MAC
112 and other peripherals. The SDRAM interface 116 may support
multiple combinations of 8, 16 or 32 bit wide SDRAMs, allowing
5 for external data storage in the range of about 2 to 32 MBytes.
The DOCSIS MAC 112 includes a number of direct memory access
(DMA) channels for fast data access to and from the system memory
114 via the internal system bus 118.

The upstream modulator 102 provides an interface with the
10 CMTS. The upstream modulator 102 may be configured to operate
with numerous modulation schemes including QPSK and 16-QAM. The
upstream modulator 102 supports bursts or continuous data,
provides forward error correction (FEC) encoding and pre-
equalization, filters and modulates the data stream and provides
15 a direct 0-65 MHz analog output.

The DOCSIS MAC 112 can also implement the upstream portions
of the DOCSIS protocol before transmission by the upstream
modulator 102. The DOCSIS MAC 112 receives data from one of the
DMA channels, requests bandwidth and frames the data for TDMA
20 with other modems on the same upstream frequency.

The DOCSIS MAC interfaces with the MIPS core 128 via the ISB
118. An exemplary embodiment of the MIPS core 128 includes a
high performance CPU operating at a speed of at least 80 MHz with
32-bit address and data paths. The MIPS core includes two way
25 set associative instruction and data caches on the order of about
4kbytes each. The MIPS core 128 can provide standard EJTAG
support with debug mode, run control, single step and software
breakpoint instruction as well as additional optional EJTAG
features.

30 The upstream modulator 102 and the downstream demodulator
100 are controlled by the MIPS core 128 via a serial interface
which is compatible with a subset of the Motorola M-Bus and the
Philips I²C bus. The interface consists of two signals, serial
data (SDA) and serial clock (SCL), which may control a plurality
35 of devices on a common bus. The addressing of the different

1 devices may be accomplished in accordance with an established
protocol on the two wire interface.

The described exemplary embodiment of the network gateway
includes a full-speed universal serial bus (USB) transceiver 1104
5 and USB MAC 122 which is compliant with the USB 1.1
specification. The USB MAC 122 provide concurrent operation of
control, bulk, isochronous and interrupt endpoints. The USB MAC
122 also can support standard USB commands as well as
class/vendor specific commands. The USB MAC 122 include
10 integrated RAM which allows flexible configuration of the device.
Two way communication of information to a device operating on a
USB can be provided, such as for example a PC on a USB 1.1
compliant twisted pair. The USB MAC 122 can be arranged for
hardware fragmentation of higher layer packets from USB packets
15 with automatic generation and detection of zero length USB
packets. The USB MAC 122 may include DMA channels which are used
to communicate received data to the system memory 114 via the
internal system bus 118. Data stored in system memory 114 may
then be processed and communicated to the cable modem termination
20 system (not shown) via the DOCSIS MAC 112 and the upstream
modulator 102. Similarly data received from the cable modem
termination system and processed by the downstream demodulator
100 and stored in system memory as higher layer packets can be
retrieved by the USB MAC122 via the ISB 118 and assembled into
25 USB packets with automatic generation of zero length USB packets.
USB packets may then be communicated to the external device
operating on the USB via the USB transceiver 104.

A media independent interface (MII) 110 can provide bi-
directional communication with devices such as for example a
30 personal computer (PC) operating on an Ethernet. The media
independent interface 110 can forward data to and receive
information from the Ethernet MAC 134. The Ethernet MAC 134 can
also perform all the physical layer interface (PHY) functions for
100BASE-TX full duplex or half-duplex Ethernet as well as
35 10BBASE-T full or half duplex. The Ethernet MAC 134 can also

decode the received data in accordance with a variety of standards such as for example 4B5b, MLT3, and Manchester decoding. The Ethernet MAC can perform clock and data recovery, stream cipher de-scrambling, and digital adaptive equalization. The Ethernet MAC 134 may include DMA channels which are used for fast data communication of processed data to the system memory 114 via the internal system bus 118. Processed data stored in system memory 114 may then be communicated to the cable modem termination system(not shown) via the upstream modulator 102. Similarly, data received from the cable modem termination system is processed by the downstream demodulator 100 and stored in system memory as higher layer packets which can then be retrieved by the Ethernet MAC 1134 via the ISB 118 and encoded into Ethernet packets for communication to the external device operating on the Ethernet via the MII 110. The Ethernet MAC 134 may also perform additional management functions such as link integrity monitoring, etc.

In addition to the SDRAM interface 116, the described exemplary embodiment of the gateway includes a 16-bit external bus interface (EBI) 140 that supports connection to flash memories 142, external SRAM 144 or EPROMS 144. Additionally, the EBI 140 may be used to interface the described exemplary network gateway with additional external peripherals. The EBI 140 can provide a 24 bit address bus and a 16-bit bi-directional data bus. Separate read and write strobes can be provided along with multiple firmware configurable chip select signals. Each chip select can be fully programmable, supporting block sizes between about 4 K-bytes and 8 M-bytes, extended clock cycle access control and 8 or 16-bit selection of peripheral data bus width. In the described embodiment, the EBI 140 can support both synchronous and asynchronous transfers. Pseudonymous transfers may be supported through the use of read/write strobes to indicate the start and duration of a transfer. The EBI 140 can include DMA access capability to or from the SDRAM interface 116. The DMA operation may take one or more forms. For example, in

operation depending on a security association to the traffic carried by a simplex connection. For example, transport mode security association between two hosts, primarily protects protocols above the IP layer while tunnel mode security association provides security and control to a tunnel of IP packets.

The exemplary security module 148 addresses possible differences in packet format between IPsec and future security applications with a generalized scheme to determine where the authentication / encryption algorithms are applied within a data packet. The authentication / encryption algorithms consider each packet to consist of three parts, a header, body and trailer. The appropriate algorithm can be applied, using any specified parameters to the body section only.

In an encryption mode, the security module 148 can add and initialize any necessary headers, determine necessary parameters, generate the associated control message and add the control and data message. In the authentication mode, the control fields of the received data packets are parsed, the parameters are determined via a security association lookup table, control message is created and the control and data messages are enqueued.

The exemplary embodiment of the network gateway includes a DMA controller 150 having a number of channels that enable direct access over the ISB 118 between peripherals and the system memory 114. With the exception of the security module 148, packets received by the network gateway 98 cause DMA transfers from a peripheral to memory, which is referred to as a receive operation. A DMA transfer from memory to a peripheral is referred to as a transmit operation. Programmable features in each channel can allow DMA controller 150 to manage maximum ISB burst lengths for each channel, enable interrupts, halt operation in each channel, and save power when certain modules are not operational. The maximum ISB burst length may be programmed independently for each channel preferably up to 64 32 bit words.

talker's voice resulting from the operation of the hybrid with its two-four wire conversion. If there is low end-to-end delay, echo from the far end is equivalent to side-tone (echo from the near-end), and therefore, not a problem. Side-tone gives users feedback as to how loud they are talking, and indeed, without side-tone, users tend to talk too loud. However, far end echo delays of more than about 10 to 30 msec significantly degrade the voice quality and are a major annoyance to the user. The audio processor can apply a fixed gain / attenuation to the conditioned signal and forwards the gain adjusted signal to the voice processor 160 via the PCM interface. In the receive mode the audio processor accepts a voice signal from the PCM interface and preferably applies a fixed gain/attenuation to the received signal. The gain adjusted signal is then interpolated from 8kHz to 96 kHz before being D/A converted for communication via a SLIC interface to a telephony device.

Each audio channel can be routed to a PCM port to allow for system level PCM testing. The PCM system tests, by way of example may require compliance with ITU G.711 for A-law and μ -law encoding / decoding.

The described exemplary embodiment of the network gateway include integrated peripherals including independent periodic interval timers 180, a dual universal asynchronous receiver-transmitter (UART) 182 that handles asynchronous serial communication, a number of internal interrupt sources 184, and a GPIO module 186 that provides multiple individually configurable input/output ports. In addition, multiple GPIO ports can be provided to drive various light emitting diodes (LEDs) and to control a number of external SLICs. A peripheral bus bridge 186 can be used to interface the low speed peripheral to the internal system bus 118.

A. DOCSIS Cable Modem

1. Downstream Demodulator

The DOCSIS downstream demodulator 100 can support 64/256 QAM. Referring to FIG. 5 the downstream demodulator 100 accepts

1 on the equalized constellation points using a quadrature
synthesizer and complex mixer under the control of the carrier
recovery loop to track out residual carrier offsets and
instantaneous phase offsets such as those caused by tuner
5 microphonics. The output of the adaptive equalizer phase
recovery block is forwarded to a forward error correction (FEC)
decoder 228. The FEC decoder can support numerous decoders
including ITU-T J.83 Annex A/B/C compatible decoders.

The Annex A/C decoder consists of four major functions,
10 frame synchronization, convolution de-interleaving, Reed-Solomon
error correction and de-randomization. Hard decisions are
preferably input into the frame synchronization block which locks
onto the inverted sync bit pattern. The MIPS core 128 sets the
characteristics of the frame synchronization acquisition and
15 retention via a CPU interface 230. Once synchronized, data
interleaving is removed by the convolution de-interleaver 232.
The de-interleaver can be based on the Ramsey III approach, and
can be programmable to provide depths from $I=1-204$ with $J=204/I$.
An on chip RAM 234 can provide for $I=1-12$. After de-interleaving
20 the data symbols are processed by a Reed-Solomon decoder, which
can correct up to eight symbol errors per RS block. The decoded
symbols are then de-randomized, which substantially undoes the
randomization inserted at the modulator. The de-randomized output
consists of MPEG-2 serial or parallel data, packet sync and a
25 data clock.

The Annex B decoder includes five layers, trellis decoding,
de-randomization, convolution de-interleaving, Reed-Solomon
decoding and checksum decoding. The Annex B concatenated coding
scheme along with interleaving provides good coding gain to
30 combat gaussian noise while still protecting against burst
errors. Soft decisions from the adaptative equalizer 226 are
input to the trellis decoder which estimates the maximum
likelihood of a sequence. The output sequences are forwarded to
a frame synchronization and de-randomization block similar to
35 those described for the Annex A/C decoders. A Reed - Solomon

1 decoder preferably corrects up to three symbol errors per RS
block. The checksum decoder accurately identifies block
containing uncorrectable errors. The downstream demodulator
outputs MPEG-2 serial or parallel data, packet sync and a data
5 clock to the DOCSIS MAC.

The downstream demodulator 100 also includes two AGC loops
which provide control for both RF and IF variable gain amplifiers
(VGAs). The gain control allocated to each loop may be
established via a CPU interface 230. The RF gain control loop
10 236 may be closed at the tuner AGC 236 while the IF loop 204 may
be completed with either an off chip VGA (not shown) or
preferably with the internal PGA 200. The power of the internal
IF signal is estimated and compared to a programmable threshold.
If the estimated power exceeds the threshold, the appropriate AGC
15 integrator is incremented by a programmable value. If the
estimated power is less than the threshold, the appropriate AGC
integrator is decremented by a comparable amount. The timing
recovery loop 206 may include a timing error discriminant, a loop
filter, and a digital timing recovery block which controls the
20 digital re-sampler. The carrier frequency/phase recovery and
tracking loops are all digital loops which simultaneously offer
a wide acquisition range and a large phase noise tracking
ability. The loops may use decision directed techniques to
estimate the angle and direction for phase/frequency
25 compensation. The loops can be filtered by integral-plus-
proportional filters, in which the integrator and linear
coefficients of the filter are programmable to provide the means
of setting the loop bandwidths. The upper bits of the loop
filter can control the direct frequency synthesizer 210,
30 providing both accurate frequency generation and fine phase
resolution.

The downstream demodulator 100 uses an on chip VCO (not
shown) referenced to a single off chip crystal which can provide
all required chip clocks. In addition, a spare D/A demodulator
35 provides a 1-bit pulse-width modulated signal which may be used

1 with an off chip RC filter. In addition, the downstream
modulator may provide tuner control ports 238 which may be used
to program two serially controlled tuner phase locked loop (PLL)
frequency synthesizers.

5 2. Upstream Modulator

Referring to FIG. 6, the upstream modulator 102 can support
QPSK and 16-QAM processing of burst or continuous data signals
102(a) received from the DOCSIS MAC. Burst encoding logic 240
includes FIFO registers and a FEC encoder, preamble preend and
10 symbol mapper. The burst FIFO register decouples the input data
rate from the transmission data rate and allows data to be input
while a burst is being actively transmitted. The FEC encoder
processes data stored in the FIFO. The FEC encoder may be a
Reed-Solomon encoder with data randomization. The parallel to
15 serial conversion of bytes entering the randomizer and serial to
parallel conversion of bits leaving the randomizer may be
programmed to be most significant bit (MSB) or least significant
bit (LSB) first. The encoder may be programmed to correct from
zero to ten symbols errors per RS block. The FEC encoder may
20 also be configured to integrate the randomizer before or after
the RS encoder. A programmable preamble of up to 1024 bits may
then be added to the data burst and the completed data burst can
be then mapped into 90 degree DQPSK, QPSK or 16-QAM.

The output of the burst encoding logic 240 is coupled to a
25 pre-equalizer 244 which may be selectively enabled to pre-distort
the transmitted waveform to offset the effects of inter-symbol
interference (ISI). The data burst is then shaped by square root
Nyquist filters 246, 248 which may have a selectable excess
bandwidth factor of 25% or 50 %. The maximum passband ripple of
30 these filters is preferably less than about 0.05 dB and the
minimum stopband attenuation is preferably greater than about 60
dB. The shaped signals are forwarded to interpolation filter
banks 250, 252 which interpolate the signal to the sample rate.
The outputs of these filters are then mixed or modulated onto
35 quadrature carriers generated by a digitally tunable frequency

1 synthesizer 258 by mixers 254, 256. The I and Q components are
then combined by summer 260. The summer 260 outputs a digital
waveform carrying the data burst whose spectrum is preferably
centered on the desired RF frequency to a D/A converter 262. The
5 D/A converter converts the digital, shaped output burst to an
analog waveform. The D/A converter 262 may have a sample rate
of up to about 200 MHz. A programmable gain attenuator 264 can
be used to provide up to about 25 dB attenuation in steps on the
order of about 0.4 dB.

10 3. DOCSIS MAC

The DOCSIS media access controller (MAC), includes baseline
privacy encryption and decryption, transmission convergence
support, a TDM/TDMA framer, and a scatter/gather DMA interface.
The transmission convergence sub-layer supports robust frame
15 acquisition and multiplexing with MPEG-TS compliant video and
audio streams. The TDM/TDMA preferably handles time
synchronization with the cable modem termination system, upstream
MAP decoding, bandwidth request generation and contention
resolution. The DOCSIS MAC may be divided into four major
20 functions, downstream functions, upstream functions, DMA
interface and miscellaneous control. The downstream functions
of the DOCSIS MAC include receiving MPEG frames 100(b) from the
downstream demodulator, extracting the DOCSIS MAC frames from the
MPEG frames, processing the MAC header, filtering messages and
25 data, processing MAP and SYNC messages, decrypting data packets
if necessary and providing cyclic redundancy checks (CRCs) on the
MAC payloads.

Referring to FIG. 7, a downstream processor 280 can include
a physical layer (PHY) interface which provides the interface to
30 the downstream demodulator (not shown). The PHY receives the
incoming MPEG stream, filters on the predefined PID, and uses the
offset value contained in the MPEG frame to find the MAC frames.
The extracted MAC frames pass to a MAC header processing block
in a message processor 282 and through a rate conversion FIFO to
35 a MAC header processing block in a downstream data encryption

5. Universal Serial Bus Transceiver and MAC

The exemplary network gateway preferably includes a USB 1.1 compliant full speed (12 M b/sec) device interface. The USB 1.1 specification defines an industry-standard USB. The specification describes the bus attributes, the protocol definition, types of transactions, bus management, and the programming interface required to design and build systems and peripherals that are compliant with this standard. The USB 1.1 specification is incorporated herein by reference as if set forth in full. The (USB) can provide a ubiquitous link that can be used across a wide range of PC-to-telephone interconnects.

The USB interface, in the described embodiment, supports sixteen configurations, four concurrent interfaces per configuration, four alternative interfaces per interface and six concurrent endpoints. An endpoint is a uniquely identifiable portion of a USB device that is the termination of a data path between the host (e.g. MIP core) and device. Each USB logical device is composed of a collection of independent endpoints. Each logical device has a unique address assigned by the system at device attachment time. Each endpoint on a device is assigned a unique device-determined identifier called the endpoint number. Each endpoint has a device-determined direction of data flow. The combination of the device address, endpoint number, and direction allows each endpoint to be uniquely referenced. Each endpoint is a simplex connection that supports data flow in one direction: either input (from device to host) or output (from host to device). An endpoint has characteristics that determine the type of transfer service required between the endpoint and the client software.

FIG. 10 shows the USB transceiver. The USB transceiver uses a differential output driver 370 to drive the USB data signal onto the USB cable in accordance with the USB 1.1 specification. The driver can be a CMOS driver with an impedance that is less than the resistance specified in USB 1.1 specification so that a discrete series resistor may be included

of data infrequently, but with bounded service periods. An interrupt transfer preferably provides a guaranteed maximum service period for the pipe as well as an attempt to re-transfer the data at the next period, in the case of occasional delivery failure due to error on the bus. The endpoint description identifies whether a given interrupt pipe's communication flow is into or out of the host.

Bulk transfers can support the exchange of relatively large amounts of data at highly variable times where the transfer can use any available bandwidth. Bulk transfers preferably provide access to the USB on a bandwidth-available basis, with guaranteed delivery of data, but no guarantee of bandwidth or latency. In addition bulk transfers preferably attempts to re-transmit in the case of delivery failure. The bulk endpoint specifies the maximum data payload size that the endpoint can accept from or transmit to the bus. This maximum applies to the data payloads of the data packets; i.e., the size specified is for the data field of the packet not including other protocol-required information. The bulk endpoint is designed to support a maximum data payload size. The bulk endpoint preferably reports in its configuration information the value for its maximum data payload size. The USB does not require that data payloads transmitted be exactly the maximum size i.e., if a data payload is less than the maximum, it does not need to be padded to the maximum size. In the described exemplary embodiment, of the USB both RX/TX bulk endpoints can support a maximum USB packet size of eight, sixteen, thirty two, or sixty four bytes. Both RX/TX isochronous endpoints 382, 383 can support a maximum USB packet size of eight, sixteen, thirty two, sixty four, one hundred and twenty eight, two hundred and fifty six or five hundred and twelve bytes. The control endpoints can support a maximum packet size of thirty two bytes and the interrupt RX interrupt endpoint can support a maximum USB packet size of eight bytes.

Both the bulk and isochronous endpoints 382, 383 can support in hardware the fragmentation of higher layer packets (such as

Ethernet packets) into USB packets in the transmit direction and the reassembly of higher layer packets from USB packets in the receive direction. An end of packet flag can be used to signal when a USB packet is shorter than the maximum packet size defined by the endpoint. In the case that the length of the higher layer packet is exactly an integer multiple of the maximum USB packet size, a zero length packet can be inserted to signal the end of packet. The USB MAC supports the generation of zero length packets in the transmit direction and the detection of zero length packets in the receive direction.

The USB MAC may include internal RX and TX RAM 384, 385 for temporary data buffering among the bulk, isochronous and control endpoints. The endpoints are then forwarded to system memory. The USB preferably includes four direct memory access (DMA) channels for fast access to and from system memory through a system bus interface 386 coupled to the ISB. Preferably, two DMA channels are used for bulk RX/TX endpoints and two are used for isochronous RX/TX endpoints.

6. Audio Processor

The audio processor module provides an interface between the voice processor and external subscriber line circuits (SLICs). Referring to FIG. 3, the audio processor includes an analog front end 172 which provides bi-directional exchange of signals over the voice bandwidth, including voice or fax/modem data signals modulated with a voice band carrier. The analog front end 172 can support four separate voice channels with an analog front end 172 having four pairs of 14-bit analog to digital converters (ADCs) and digital to analog converters (DACs).

FIG. 11 shows a block diagram of the analog front end of the audio processor. The digital input/output data of the DAC/ADC can be interpolated / decimated in the codec DSP logic block to provide 14-bit, 8 kHz input/output data for the audio processor 170. A pair of resistors 391a, 391b at the output of each DAC 390 converts the current output signal to a voltage. A pair of

1 filter 425 is coupled to the echo canceller input to match the
echo path impulse response.

Programmable gain adjuster 428 applies a fixed gain or
attenuation to output 426(a) of the difference operator 426.
5 Gain adjuster 428 can provide programmable gain / attenuation
adjustments of +/- 20 dB with step size of 1 dB. A fixed gain
of attenuation 429 is applied to the gain adjusted signal. A
multiplexer 427 coupled to the output of the fixed gain 429
allows the signal to be routed to a A-law / μ -law (G.711
10 compatible) encoder 430 which is coupled to an external PCM port
which allows for external testing of the audio processor module.
Multiplexer 429 also forwards the gain adjusted output signal to
the voice processor 160 via the DSP interface 168 (see FIG. 3).

The described exemplary embodiment of the audio processor
15 includes multiplexer 431 coupled to the data interface in the
receive mode. Multiplexer 431 may couple decoded samples to a
A-law / μ -law decoder 432 which is also coupled to an external
PCM port to allow for external testing of the audio processor
module. The multiplexer 431 may also forward decoded samples
20 from the data interface to a gain adjuster 435 which applies a
fixed gain or attenuation to the decoded signal 433. Gain
adjuster 435 compensates for system level gain adjustments and
may provide programmable gain/attenuation adjustments on the
order of about +/- 20dB with a step size of 1 dB. A 1 kHz test
25 tone generator 434 that provides a digital representation of a
1004 Hz test tone at a level of 0 dBm. The test tone may be
optionally injected into the data stream by summer 436 to debug
and verify the audio processor. The test tone may be
configurable in both frequency and amplitude, although it is
30 preferably limited by the 8 kHz sample rate such that only 0-4
kHz may be allowed.

An interpolater 438 modifies the sample rate from 8 to 96
kHz. The interpolater 438 can be implemented with a FIR filter
which may be either minimum phase or linear phase. A minimum
35 phase filter is preferred for voice applications which require

low group delay but may tolerate group delay distortion which may be introduced by the minimum phase filter. A linear phase filter is preferred for fax and or modem applications. In addition, a metering pulse generator 440 can be used to generate 12/16 kHz metering pulses that are summed with the interpolated signal by summer 442. The metering pulses allow a user to monitor the cost of the call as it progresses. The rate at which the metering pulse generator 440 transmits the metering pulses varies depending on the rate per minute of the call. The minimum gap between pulses is, by way of example, on the order of about 100 msec but may be much greater for inexpensive local calls. The amplitude of the metered pulses can be adjustable to accommodate impedance changes during the on hook, off hook and ringing states. The interpolated signals are forwarded to the DAC 390 for communication to a telephony device via the SLIC (not shown).

Power efficiency is often important for telephony applications. Therefore, the described exemplary embodiment of the audio processor includes the capability to enter a power saving/sleep mode wherein only those functions that are required for wake up are active. All circuits should be capable of reaching active steady state within about a 10 msec activation period.

The described exemplary embodiment of the preferred audio processor 170 further includes a ring generator which synthesizes reference waveforms which are forwarded to the SLIC to ring telephony devices. The ring generator can be used to support sinusoidal, trapezoidal and square waves. In addition the ring generator frequency, amplitude and DC offset can be configurable. A block diagram of a preferred ring generator 450 is shown in FIG. 13. The ring generator 450 includes a digital oscillator 452 which generates a sinusoid of a given frequency and amplitude determined by instructions 454, 456 from the DSP core of the voice processor 160 via the DSP/PB interface 168 (see FIG. 3). The sample rate of the sinusoid can be, by way of example, on the order of about 1000 Hz, divided down from the 24.576 MHz

system clock input 458. A variable clipper 460 symmetrically clips the sinusoid about zero such that the sinusoid may be converted into either a trapezoid or into a square wave. The DSP core of the voice processor 160 (see FIG. 3) can be used to define the thresholds 462 with which the sinusoidal waveform is clipped. The clipped waveform can be scaled by multiplier 464, which applies a signal attenuation 466 defined by the voice processor 160. Summer 468 provides a configurable DC offset by adding a DC bias 470 as established by the voice processor. The offset may vary from negative full scale to positive full scale. A converter 472 can be used to convert the ring waveform 468(a) into a single bit representation. A single pole analog filter may be included on the output of the converter to reduce the quantization noise generated by the converter 472. The filtered signal is then forwarded the analog front end 172 for communication to a telephony device via the SLIC. In the described exemplary embodiment, of the audio processor each audio channel may be routed to a PCM port to allow for system level PCM testing. The PCM system tests, by way of example, can require compliance with ITU G.711 for A-law and μ -law encoding / decoding.

A. Voice Synchronization

Digitizing and transmitting voice data via packets in a network system is a common telephony problem. Customarily Pulse Code Modulation (PCM) techniques digitize a voice signal by sampling an analog voice signal and converting each sample into a digital code which is communicated between compatible receiving and transmitting systems on the near and far ends of the network. In addition, in a voice band data mode, the exemplary network gateway may transparently exchange data without modification (other than packetization) between a near end telephony device (or circuit switched network) and the packet based network. This is typically used for the exchange of fax and modem data when bandwidth concerns are minimal. The problem that arises is that the receiving system's clock may not be correlated with the

1 transmitter's clock. This difference, even if minute, may cause
the sample buffer in the receiving unit to underflow or overflow.
In the case of data underflow, samples are extracted from a
sample buffer faster than samples are written to the buffer so
5 that the system may collapse from data starvation. During data
overflow, voice signals transmitted from one communication port
enter the sample buffer in the receiving unit faster than they
are extracted. The resulting overflow of data signals may result
in artifacts in a voice call or data in voiceband data mode.

10 To prevent data signal overflow and underflow, it is,
therefore, desirable to synchronize the receiving clock to the
incoming data rate. A voice synchronizer may be used for this
purpose. Although the the voice synchronizer is described in
the context of an audio processor system within a network
15 gateway, those skilled in the art will appreciate that the voice
synchronizer is likewise suitable for various other telephony and
telecommunications application.

Referring to FIG. 14, network gateway 480 supports the
exchange of voice between a hybrid fiber coaxial (HFC) network
20 482 and a traditional circuit switched 484 or a packet based
network 486. In an exemplary embodiment, telephony device 490
is connected to the PSTN over PSTN telephone gateway 492. The
PSTN telephone gateway 492 may be clocked by a telephony network
clock signal 494(a) from network clock reference 494 which is
25 also coupled to CMTS 496 such that the PSTN telephone gateway 492
may be synchronized with the CMTS clock for the transfer of PCM
voice packets 492(a) between the CMTS 496 and the PSTN telephone
gateway 492. The telephony network clock is preferably a
conventional Building Integrated Timing Supply (BITS) clock. The
30 equipment requirements for interfacing to this clock are known
to those skilled in the art and are described in Bellcore
document TR-NWT-001244 the content of which is incorporated
herein by reference as if set forth in full. The CMTS clock is
synchronized with the telephony network clock signal 494(b) via
35 CMTS clock synchronizer 498 which utilizes headend reference tick

00E F e t " 5 4 7 2 2 6 0

1 clock 500, as described in U.S. Pat. Application No. 09/501,850,
the contents of which is incorporated herein by reference as if
set forth in full.

5 A timebase message generator 502 generates time messages or
stamps which provide the network gateway 480 an absolute time
reference which is representative of the frequency of the crystal
oscillator timing reference (not shown) of the CMTS 496, which
is in turn synchronized to the telephone network clock 494.

10 Referring to FIG. 14A, the network gateway 480 preferably
includes a digital tracking loop 504 that provides a
substantially stable clock output 510(a). A difference between
the time stamps forwarded by the CMTS and the output of a local
time reference, that is derived from a numerically controlled
15 oscillator 508, is formed by differencing circuit 510. This
difference defines a frequency error value. This frequency error
value is filtered by loop averaging filter 511 which prevents
undesirable deviations in the frequency error value from
affecting the numerically controlled oscillator 508 in a manner
which would decrease the stability thereof or cause the
20 numerically controlled oscillator 508 to operate at other than
the desired frequency. The loop filter 511 can be configured so
as to facilitate the rapid acquisition of the frequency error
value, despite the frequency error value being large, and then
to reject comparatively large frequency error values as the
25 digital tracking loop 504 converges, i.e., as the output of the
local timing reference becomes nearly equal to the absolute time
reference, thereby causing the frequency error value to approach
zero.

30 Referring to FIG. 15, in the downstream direction, voice or
voiceband data packets are received by the network gateway 480,
demodulated and re-formatted by the downstream demodulator /
DOCSIS MAC 512 and forwarded to system memory 514. In this and
other embodiments, common reference numerals may be used to
represent like components. The voice data is forwarded to a
35 decoder system (not shown) within the voice processor 516. The

decoder system preferably provides delay compensation, voice decoding, DTMF generation, call progress tone generation, comfort noise generation and lost frame recovery. Processed voice sample are then forwarded to a first voice queue 518. A voice synchronizer 520 is coupled to the output of the first voice queue 518. The voice synchronizer 520 re-samples the voice frames stored in the first voice queue 518 in accordance with an error signal and forwards re-sampled voice signals to a second voice queue 522 so that the rate at which samples are removed from the second voice queue 522 by a DAC 524 matches the rate at which they are inserted into the second voice queue 522 by the voice synchronizer 520.

In operation, each time the clock of the DAC 524 decrements to zero, a sample can be removed from the second voice queue 522 and transmitted to the near end telephony device 526 via a subscriber line interface circuit 525. In the described exemplary embodiment, the DAC 524 is preferably driven by sampled DAC clock 528. In a jitter-free system, the DAC 524 removes frames from the second voice queue 522 at the exact same rate at which frames are inserted into the first voice queue 518 by the voice processor 516. However, when jitter or other impairments are present voice synchronization is needed because the DAC clock of the receive unit within the network gateway may not be correlated to the far end sample clock that generated the received data. In a data underflow condition in the downstream direction, the DAC clock 528 in the network gateway 480 leads the far end sample clock so that if left uncorrected samples would be removed from the second voice queue 522 faster than they are being inserted in the first voice queue 516, so that the system may collapse from data starvation. During a data overflow condition in the downstream direction, the DAC clock 528 in the network gateway lags the far end sample clock so that samples are inserted into the voice queue faster than they are removed. The resulting overflow may result in artifacts in a voice call or data in voiceband data mode.

3072, the incoming signal is upsampled by the filtered counter output 543(a) divided by 3072 ($A/3072$). Similarly, when filtered counter output 543(a) is less than 3072, the incoming signal is down-sampled by the filtered counter output 543(a) divided by 3072 ($A/3072$).

Referring to FIG. 18B, in an alternate voice synchronizer in the downstream direction, the MIPS core 513 increments a counter 553 when the network gateway 480 receives a new voice frame from the CMTS (not shown). The counter 553 preferably utilizes the DAC 24.576 MHz clock 528 as a reference. The counter output 553(a) provides the difference between the frame clock and the DAC clock. A low pass filter 543' again smooths or averages the difference in various frame arrival rates as previously described. The sample rate tracker 548 fractionally decimates or interpolates the incoming signal 548(a) in accordance with the output of filtered counter count 543(a) as compared to 24576. In the case where the frame arrival clock and the DAC clock are perfectly correlated the counter 553 would output 24576 and the incoming signal would not be resampled. However, for the case where the filtered counter output 543(a) is greater than 24576 the sample tracker 548 upsamples the incoming signal by the output 543(a) of the low pass filter divided by 24576 (i.e. $A/24576$). Similarly, when the filtered counter output 543(a) is less than 24576, the sample rate tracker 548 down-samples the incoming signal by the output 543(a) of the low pass filter divided by 24576 ($A/24576$).

Referring to FIG. 18C, an alternate error generation unit utilizes packet control information 554 to generate a far end clock error signal 555 representing the difference between the far end sampling rate and the near end voice playout sampling rate. The clock error signal may then be used to fractionally resample the received data packets 556. In operation the DSP of the voice processor module may read from and write to data packet control registers within local memory of the DSP to monitor and control the flow of downstream data. For example, referring back

1 will appreciate that increasing the frequency of the TRC clock
536 will provide increasingly accurate error estimates. The
first and second latches 560, 561 coupled to counters 558, 559
respectively are responsive to packet arrival control signals 562
5 and packet complete control signals 563 respectively. Latch 560
therefore outputs the count value between successive packet
arrival control signals, i.e. last count 560(a) and current count
560(b) and latch 561 outputs the count value between successive
packet complete control signals, i.e. last count 561(a) and
10 current count 561(b). Difference operator 564 is coupled to
latch 560 and outputs the difference between the current and the
last packet arrival counts, i.e. the count between successive
packet arrival control signals output by latch 560. Similarly,
difference operator 564a is coupled to latch 561 and outputs the
15 difference between the current and the last packet complete
counts, i.e. the count between successive packet complete control
signals output by latch 561. Low pass filters 543 and 543' are
coupled to difference operators 564, 564a respectively. The low
pass filters 543 and 543' as previously described may be single
20 pole filters that smooth the transitions between different
sampling rates.

A difference operator 565 is coupled to the output of low
pass filters 543 and 543' and forwards an estimate of the
difference between the packet arrival period and packet complete
25 period to the sample rate tracker 534. The sample rate tracker
also receives the incoming PCM voice data. The sample rate
tracker 534 fractionally decimates or interpolates the incoming
signal 534(a) in accordance with the ratio of the packet arrival
count and the packet complete count. Thus if in operation the
30 packet arrival count is larger than the packet complete count,
the sample rate tracker up-samples the incoming signal 534(a) by
the ratio of the packet arrival count to the packet complete
count. Otherwise the sample rate tracker 534 downsamples the
incoming signal 534(a) by the ratio of the packet arrival count
35 to the packet complete count. The sampling rate tracker 534

0974E4300

1 signal 575(b) from the near end digital input signal 572(a) in
a difference operator 576. Typically, the adaptive filter 575
converges or adapts only in the absence of near end speech.
Therefore, near end speech and/or noise present on the near end
5 digital input signal 572(a), typically referred to as the double
talk condition, may cause the adaptive filter 575 to diverge.
Traditionally, echo cancellers utilize energy estimators
577a,577b to estimate the energy (E_{near}) of the near end signal
572(a) and the energy (E_{far}) of the far end 575(a) signal. A
10 typical double algorithm 576 then declares near end speech
active, disabling adaptation of the adaptive filter 575, when the
energy of the near end signal is greater than the energy of the
far end signal times the hybrid gain(H), ($E_{near} > H * E_{far}$).

A primary disadvantage of conventional approaches which
15 utilize energy estimates is the delay introduced into the
detection of near end speech by the energy estimators (typically
low pass filters) which may significantly corrupt the output of
difference operator 576, which is typically used as a the
feedback error for filter adaptation. The described exemplary
20 echo canceller includes a double talk algorithm that provides
rapid detection of near end speech in the presence of far end
speech along with a look ahead capability so that the adaptive
filter may halt adaptation (i.e. freeze the filter taps or
coefficients) before the near end speech reaches the difference
25 operator.

Although echo cancellation is described in the context of
an audio processor for voice exchange via a network gateway,
those skilled in the art will appreciate that the techniques
described for echo cancellation are likewise suitable for various
30 applications requiring the cancellation of reflections, or other
undesirable signals, from a transmission line. Accordingly, the
described exemplary embodiment for echo cancellation in a signal
processing system is by way of example only and not by way of
limitation.

35

Referring to FIG. 20, a high pass filter 587 receives a reference signal 587(a). The high pass filter 587 matches the echo path impulse response of the Rx data line. The output of the high pass filter 587 is input into the adaptive filter 575 that models the transfer characteristics of the dialed telephone line circuit. The adaptive filter 575 may be a linear transversal filter or other suitable finite impulse response filter. In addition, Rx data from the far end 581(a) is coupled to double talk logic 580 before the interpolator 581 of the audio processor (not shown) and the DAC 588 of the analog front end (not shown). The double talk logic 580 therefore receives a far end reference signal $F(n)$ 580(a) with an 8 kHz sampling rate. In addition, the double talk logic 580 is preferably coupled between the ADC 589 of the analog front end and the decimator 582 of the audio processor (not shown). A downsampler 583 performs 12:1 sample decimation of the 96kHz near end Tx data 572(a) and forwards the decimated near end data samples 583(a) to the double talk logic at an 8 kHz sample rate. To minimize delay, the downsampler does not low pass filter the near end samples 572(a) prior to decimation. Aliasing components which may be created are insignificant in that the output of the downsampler 583(a) simply drives the double talk detection logic 580 and is not transmitted to the far end. An energy estimator 584 estimates the background noise level of the decimated near end signal 583(a) and forwards the estimated level to the double talk logic 580. The energy estimator 584 is preferably a low pass filter with a long time constant, on the order of about 10 seconds. With a long time constant the energy estimator tends to track the minimum energy level of the decimated near end signal 583(a). Energy estimator 585 estimates the short term energy of the far end TX data $F(n)$.

The adaptive filter 575 can be based upon a normalized least mean square algorithm (NLMS) as described in S. Haykin, *Adaptive Filter Theory*, and T. Parsons, *Voice and Speech Processing*, the contents of which are incorporated herein by reference as if set

1 forth in full. An error signal 576(a) at the output of the
difference operator 576 for the filter adaptation may be
characterized as follows:

5
$$e(n) = Tx(n) - \sum_{j=0}^{L-1} w(j)F(n-j)$$

where $e(n)$ is the error signal at time n , $F(n)$ is the
reference signal 587(a) at time n and $Tx(n)$ is the Tx data signal
586(a) input into the difference operator 576 at time n , and $w(j)$
10 are the coefficients of the transversal filter where the
dimension of the transversal filter is the worst case echo path
length (i.e. the length of the tail circuit L) and $W(j)$, for $j=0$
to $L-1$, is given by:

15
$$w(j) = w(j) + \mu * e(n) * F(n-j)$$

wherein $w(j)$ is preferably initialized to a reasonable value
such as for example zero.

Assuming a block size of four msec (or 32 samples at a
20 sampling rate of 8 kHz), the short term average energy of the
reference signal E_{ref} is the sum of the last 32 reference samples
so that the convergence gain may be given by:

25
$$\mu = \frac{\alpha}{E_{ref}(n)}$$

where α is the adaptation step size and E_{ref} is the energy
estimate of the far end data sample $F(n)$. In the described
exemplary embodiment α , is set to zero when near end voice is
detected so that the convergence gain μ is equal to zero and the
30 filter coefficients are not updated. Otherwise α is set to a
constant of less than one and preferably in the range of 0.8-
0.95. One of skill in the art will appreciate that the adaptive
filter may be implemented in a variety of ways, including fixed
point rather than the described floating point realization.

1 Accordingly, the described exemplary adaptation logic is by way of example only and not by way of limitation.

5 The 96 kHz near end Tx data samples 572(a) are also decimated by a second 12:1 decimator 582. However, decimator 582 does include a low pass filter capability to prevent aliasing of the decimated signal. The decimated output 582(a) is forwarded to a 60 Hz high pass filter 586 which reduces the 60 Hz interference induced on the transmit line due to proximity to power lines. Filtered output 586(a) is input to the difference
10 operator 576 that preferably cancels unwanted echo by subtracting filtered reference signal 575(b) from the filter output signal 586(a).

15 In the described exemplary embodiment, the adaptive filter 575 models the transfer characteristics of the hybrid and the tail circuit of the telephone circuit. The tail length supported should preferably be at least 8 msec. The adaptive filter 575 may be a linear transversal filter or other suitable finite impulse response filter. The echo canceller preferably converges or adapts only in the absence of near end speech. Therefore,
20 near end speech and/or noise present on the input signal 572(a) may cause the adaptive filter 575 to diverge. To avoid divergence, the adaptive filter 575 can be selectively enabled by the double talk logic 580. The double talk logic 580 utilizes a sample based algorithm to detect the presence of near end
25 speech without incurring the delays associated with conventional systems in accordance with the following equation:

$$| \text{Near} | > H * \text{Peak} \{ | F(n) | \} + \text{Background_Noise}(n)$$

30 The double talk logic 580 is used to declare near end speech active when the absolute value of the decimated near end signal 583(a) is greater than the product of the hybrid gain (H) and a peak statistic of the far end data samples 581(a) summed with the estimated background noise of the transmit data samples. The hybrid gain is generally a constant preferably less than about one-half. The background noise for a typical voice channel is
35 on the order of about -70 dBm which is far less than average

active speech levels, which are typically in the order of about -25 dBm. The background noise estimate is therefore initialized to a value of about -70 dBm and thereafter periodically updated 584(a) by the energy estimator 584. The peak statistic of the far end data samples is defined by the following logic:

```
If max{A * [| F(n) |, ..., | F(n-L-1) |]} > Peak(n-1) then
Peak(n) = max {A [ |F(n)|, ..., * |F(n-L-1)|]}
else
Peak(n) = d * Peak(n-1);
```

where A is a weighting function that is greater than zero and less than or equal to one. The parameter L is the number samples over which the maximum is determined, typically in the range of zero to one hundred and twenty eight samples and preferably on the order of about 64 samples. The parameter d is preferably a constant that is also greater than zero and less than or equal to one and preferably on the order of about 0.99. Therefore, to determined the peak statistic of the far end, the double talk logic applies a weighting factor A to the absolute value of the current sample (F(n)) and previous L samples (F(n-L)). If the maximum product is greater than the previous peak statistic Peak(n-1) then the current peak statistic Peak(n) is set at the maximum of the product of the weighting factor and far end samples. Otherwise the current peak statistic Peak(n) is set equal to d times the value of the previous peak statistic Peak(n-1).

In the described exemplary embodiment, A, L and d are empirically determined to optimize performance and computational load for a given application. For example, double logic 580 can more accurately detect near end speech if the maximum is determined over a larger number of samples L. However, computational intensity also increases with increasing number of samples L. A and d can be inversely related to the number of sample L, so that A and d are smaller for larger number of samples and vice versa.

1 In the described exemplary embodiment, there is a delay
associated with the high-pass filter 586 and the decimator 582.
The double talk logic 580, which has negligible delays, can
receive and process near end data samples prior to their arrival
5 at the difference operator 576. Thus, the delay associated with
the high-pass filter 586 and the second decimator 582 provide a
look-ahead of M samples allowing the double talk logic 580 to
preferably disable adaptation of the adaptive filter 575 M
samples before the near-end signal reaches the difference
10 operator 56. The look ahead capability M is the sum of the
equivalent delays associated with the high pass filter 586 and
the second decimator 582 and is typically two-three 8kHz samples
for a ITU-T G712 compliant system.

FIG. 20A shows another approach for echo cancellation where
15 the near end digital signal after decimation to an 8 kHz signal
582(a) is input to the double talk logic 580. This approach can
be utilized in systems where the echo canceller and codec are not
integrated so that the near end data samples have previously been
decimated. In this instance, a look ahead buffer 588 receives,
20 buffers, and forwards decimated near end signals 582(a) to the
difference operator 576, providing a look ahead capability of M
samples where M may be optimized for a given application to
balance performance, computational intensity and delay.

The relative strength of the near end signal compared to the
25 echo coupled through the hybrid increases with decreasing hybrid
gain (H) so that in the described exemplary embodiment, the
accuracy of near end voice detection increases with decreasing
hybrid gain(H). Referring to FIG. 21, in another aspect of the
present invention, a short adaptive filter 590 is integrated into
30 the preferred double talk detection algorithm. The adaptive
filter 590 models the transfer characteristics of the dialed
telephone line circuit. The adaptive filter 590 may be a linear
transversal filter or other suitable finite impulse response
filter. An error signal 591(a) at the output of the difference
35

operator 591 for filter adaptation may be characterized as follows:

$$e_0(n) = Tx_0(n) - \sum_{j=0}^{K-1} w_0(j)F(n-j)$$

where $e_0(n)$ is the error signal at time n 591(a), $F(n)$ is the reference signal 580(a) at time n and $T_{x_0}(n)$ is the Tx data signal 591(b) input into difference operator 591 at time n , and $w_0(j)$ are the coefficients of the transversal filter where the dimension of the transversal filter is preferably the worst case echo path length (i.e. the length of the tail circuit K) and $w_0(j)$, for $j=0$ to $K-1$, is given by:

$$w_0(j) = w_0(j) + \mu * e_0(n) * F(n-j)$$

wherein $w_0(j)$ is preferably initialized to a reasonable value such as for example zero.

Assuming a block size of one msec (or 8 samples at a sampling rate of 8 kHz), the short term average energy of the reference signal E_{ref} is the sum of the last eight samples so that the convergence gain may be given by:

$$\mu = \frac{\alpha}{E_{ref}(n)}$$

where α is the adaptation step size and E_{ref} is the energy estimate of the far end data sample $F(n)$. In the described exemplary embodiment, the double talk logic 580 does not selectively enable / disable adaptation of the filter 590 in accordance with the detection of near end speech so that filter 590 continuously adapts. Therefore, to reduce the computational burden placed upon the system and to prevent the filter from diverging the adaptive filter 590 can be figured to adapt very slowly so that α is preferably in the range of about 0.01 - 0.0001.

1 The adaptive filter 590 again filters the far end reference
signal 581(a) so that the echo level is can be reduced by
subtracting filtered reference signal 590(b) from the Tx data
samples 591(b) in a difference operator 591. The adaptive filter
5 590 can be reduce line echos about 6-12 dB so as to improve the
performance of the double talk logic. In the described exemplary
embodiment, the output 591(a) of the difference operator 591,
(i.e. Tx data samples with reduced echo) is then forwarded to
double talk logic 580 which then detects near end speech and
10 selectively enables/disables adaptation of the adaptive filter
575(see FIG. 20).

7. Voice Processor

00446600
15 The Internet is a loose association of thousands of networks
and millions of computers across the world that are
interconnected through communication links. The emergence of
Internet Protocol (IP) as the standard transport protocol for
packet based networks has enabled an on-line revolution in
communications service and applications. Traditional dial-up
modems provide online access through the public telephone network
20 at up to 56 Kbps (equal to 56,000 bits per second). A cable
modem, on the other hand, provides users with high-speed Internet
access through a cable television network at data rates as high
as 56 Mbps. However, traditional cable modem service has been
limited to data applications so that the realization of diverse
25 communications services at increased data rates requires the
development of a common broadband cable access network with
integrated voice and data services. Cable Television
Laboratories, Inc. (CableLabs®) a membership organization
consisting of cable television system operators developed
30 PacketCable 1.0 which defines interface specifications for
interoperable equipment capable of providing packet-based voice,
video and other high-speed multimedia services over hybrid fiber
coax (HFC) cable systems utilizing the DOCSIS protocol.

35 The described exemplary network gateway includes a voice and
data processor that supports the exchange of voice and data

1 between a traditional circuit switched and a packet based network
via a DOCSIS HFC network. The exemplary voice and data processor
may be implemented with a programmable DSP software architecture
as shown in FIG. 22. This architecture includes a high speed DSP
5 600 with program memory 602, preferably on the order of about a
80k word SRAM, and data memory 604 preferably on the order of
about a 48k word SRAM. A PCM highway 606 provides the voice and
data processor 160 access to the audio processor and optional
external audio processing circuits. A grant synchronizer 608
10 insures delivery of samples to the network gateway for upstream
transmission. The grant synchronizer signals the DSP 600 that
a pending grant is about to arrive at the network gateway so as
to allow the DSP 600 to synchronize itself to scheduled grants
at the network gateway. A host interface 610 transfers data,
15 control and status messages between the DSP 600 and the MIPS core
128.

The described exemplary embodiment preferably provides
embedded media terminal adapter (MTA) capability in compliance
with PacketCable 1.0. The exemplary embedded MTA may be
20 implemented with the programmable DSP software architecture to
provide a subscriber side interface to the subscriber's telephony
device via the voice and data processor, as well as a network
side interface to the DOCSIS cable modem. Referring to FIG. 23
the preferred embedded MTA 620 includes a host application
25 programming interface (HAPI) 621 that provides a software
messaging interface between the MIPS host and the voice and data
processor DSP. The HAPI 621 facilitates the issuing of commands
from the MIPS host to the voice and data processor DSP as well
the sending of events from the DSP to the MIPS core host.

30 In addition, the MTA 620 can provide all signaling and
encapsulation elements required to provide telephony service over
a DOCSIS HFC network 622 including media transport and call
signaling via quality service logic 623. For example, gateway
control protocol (GCP) logic 624 receives and mediates call-
35 signaling information between the PacketCable network and the

When a voice channel is successfully established, real time transport protocol (RTP) is used to transport all media streams in a PacketCable compliant network to guarantee interoperability. Real time transport protocol (RTP) provides end-to-end delivery services for data with real time characteristics, such as interactive audio and video. Those services include payload type identification, sequence numbering, timestamping and delivery monitoring of the quality of service (QoS) and conveys to participants statistics such as for example packet and byte counts for the session. RTP resides right above the transport layer. The described exemplary embedded MTA 620 preferably includes RTP logic 630 that converts RTP packets (headers) to a protocol independent format utilized by the voice and data processor 626 and vice versa.

The described exemplary embedded MTA preferably includes channel associated signaling (CAS) logic 632 resident on the MIPS core that interfaces with the subscriber line interface circuits 634 via the GPIO interface 184 (see FIG. 3) to provide ring generation, hookswitch detection, and battery voltage control. The CAS logic 632 preferably supports custom calling features such as for exam distinctive ringing.

The described exemplary embedded MTA 620 preferably includes MTA device provisioning logic 636 which enables the embedded MTA 620 to register and provide subscriber services over the HFC network 622. Provisioning logic 636 provides initialization, authentication, and registration functions. The Provisioning logic 636 also provides attribute definitions required in the MTA configuration file. The provisioning logic 636 includes a SNMP logic 638 that exchanges device information and endpoint information between the MTA 620 and an external control element called a provisioning server (not shown). The MTA also sends notification to the provisioning server that provisioning has been completed along with a pass/fail status using the SNMP protocol.

1 The Provisioning logic 636 also includes DHCP logic 640
which interfaces with an external dynamic host configuration
protocol (DHCP) server to assign an IP address to the MTA. The
DHCP server (not shown) is a back office network element used
5 during the MTA device provisioning process to dynamically
allocate IP addresses and other client configuration information.
Further provisioning logic preferably includes domain name server
(DNS) logic 642 which interfaces with an external DNS server(not
shown) to obtain the IP address of a PacketCable server given its
10 fully\qualified domain name.

 The MTA configuration file is downloaded to the MTA from an
external trivial file transfer protocol (TFTP) server (not shown)
through TFTP logic 644. The TFTP server is a back office network
element used during the MTA device provisioning process to
15 download configuration files to the MTA. An HTTP Server may be
used instead of a TFTP server to download configuration files to
the MTA.

 Each of PacketCable's protocol interfaces is subject to
threats that could pose security risks to both the subscriber and
service provider. The PacketCable architecture addresses these
20 threats by specifying, for each defined protocol interface, the
underlying security mechanisms (such as IPSec) that provide the
protocol interface with the security services it requires, e.g.,
authentication, integrity, confidentiality. Security logic 646
25 is PacketCable compliant and provides for voice and provides
end-to-end encryption of RTP media streams and signaling
messages, to reduce the threat of unauthorized interception of
communications. The security logic 646 preferably provides
additional security services such as, for example,
30 authentication, access control, integrity, confidentiality and
non-repudiation.

 DOCSIS service logic 648 preferably provides the primary
interface between the MTA 620 and the DOCSIS cable modem (i.e.
DOCSIS MAC and modulator / demodulator) of the network gateway.
35 The DOCIS service logic 648 provides multiple sub-interfaces such

1 as for example a control sub-interface which manages DOCSIS
service-flows and associated QoS traffic parameters and
classification rules as well as a synchronization interface which
is used to synchronize packet and scheduling prioritization for
5 minimization of latency and jitter with guaranteed minimum
constant bit rate scheduling. In addition, the DOCSIS service
logic is used to request bandwidth and QoS resources related to
the bandwidth. The DOCSIS cable modem features of the network
gateway then negotiate reserve bandwidth, guaranteed minimum bit
10 rate etc, utilizing DOCSIS 1.1 quality of service feature.
Similarly, DOCSIS service logic 648 preferably includes a
transport interface which is used to process packets in the media
stream and perform appropriate per-packet QoS processing.

The exemplary embedded MTA may best be illustrated in the
15 context of a typical voice communication across the DOCSIS HFC
network. The user initiates a communication by going off hook.
The CAS detects the off hook condition from the SLIC and sends
an off hook event to the MTA call client. The MTA call client
then instructs the GCP logic to generate a off hook signal. The
20 GCP logic generates an of hook signal which is forwarded to the
MTA call client and transmitted out the QoS service logic to the
call management server via the DOCSIS MAC and upstream modulator
of the network gateway and the CMTS. The call management server
typically would transmit a return signal via the CMTS, DOCSIS MAC
25 and downstream demodulator of the network gateway to the MTA call
client via the QoS service logic. The MTA call client preferably
forwards that signal to the GCP logic which decodes the signal,
typically play dial tone. The GCP logic would then signal the
MTA call client to play dial tone. The MTA call client then
30 sends a command to the voice and data processor via the HAPI
interface to play dial tone. The user then hears a dial tone.

Upon hearing a dial tone a user will then typically dial a
number. The voice and data processor includes a DTMF detector
which detects the dialed digits and forwards the detected digits
35 to the MTA call client as events via the HAPI interface. The MTA

call client forwards the event to the GCP logic which encodes the dialed digits into a signaling message which is returned to the MTA call client. The MTA call client transmits the signaling message out the QoS service logic to the call management server via the DOCSIS MAC and upstream modulator of the network gateway and the CMTS. The call management server would then instruct a called party MTA to generate a ring to the called number. If the called number answers by going off hook, the CAS of the called MTA would detect an off hook condition and signal the call management server. The call management server then instructs the MTA call client via the CMTS, and downstream demodulator, DOCSIS MAC and QoS service logic of the network gateway to establish a voice connection with a given set of features, i.e. use echo cancellation, and silence suppression, use given coder etc. In addition, the MTA call client is given the IP address of the called party, to which the RTP voice packets should be sent. The MTA call client forwards the received message to the GCP logic which decodes the received message. The GCP logic generates attribute instructions for the voice and data processor such as, for example, encoding method, use of echo cancellation, security parameters, etc. which are communicated to the voice and data processor via the MTA call client and the HAPI interface.

Voice packets are then exchanged. For example, if the calling party speaks, the voice and data processor would process the voice and forward voice packets the MTA call client via the HAPI interface. The MTA call client would then forward those voice packet the RTP logic which would convert the packet from a protocol independent packet format to the RTP format. The RTP voice packets are then returned to the MTA which transmits the RTP voice packet to the CMTS via the QoS service logic and the DOCSIS MAC and upstream demodulator of the network gateway. The voice packets are then routed to the called party. Similarly, voice packets from the called party are communicated to the MTA of the call client via the QoS service logic. The MTA call client forwards the RTP voice packets to the RTP logic which

1 converts the packet from the RTP format to the protocol
independent packet format. The protocol independent voice
packets are returned to the MTA call client which forwards them
to the voice and data processor via the HAPI interface. The
5 voice and data processor decodes the packets and communicates a
digital stream to the called party. Voice exchange would continue
in a similar manner until an on hook condition is detected by
either the calling or called party CAS which would forward a
on hook detection event to its respective MTA. The MTA would
10 instructs the GCP logic to generate a hook detection signaling
message which is returned to the MTA and forwarded to the call
management server. The call management server would generate a
request to play (dial tone, silence or receiver off hook) which
is forwarded to the opposite MTA. The MTA would forward the
15 request to the GCP logic which would then instruct the voice and
data processor to play dial tone via the MTA and HAPI interface.

Telephony calls in the other direction are similarly
processed. For example, the call management server instructs the
MTA called client to ring a dialed number. The MTA called client
20 instructs the GCP logic to generate a command to ring the
dialed number. The command is then forwarded to the CAS via the
MTA called client. The CAS generates a ring signal and forwards
that signal to the SLIC which then rings the called telephony
device. The MTA called client may also instruct the GCP logic
25 to present call ID which preferably generates a command for the
voice and data processor to present caller ID. If the user picks
up the phone the CAS would detect an off hook condition and
signal an off hook event back to the MTA. The MTA called client
would then instruct the GCP logic to create an off hook detection
30 signaling message, which when created is returned to the MTA and
forwarded to the external call management server via the QoS
service logic, DOCSIS MAC and upstream modulator of the network
gateway and the CMTS. A communication channel would again be
established with a given set of attributes as previously
35 described.

1 signal processing software from the underlying hardware. This methodology allows the software to be ported to various hardware platforms by porting only the hardware interface portions of the HAPI interface 621 to the target hardware. The physical
5 interface 708 formats the message in accordance with the underlying DSP and forwards or transmits the message to the telephony algorithms executing on the DSP.

Similarly, the underlying DSP forwards processed packets, such as, for example, encoded voice packets, to the physical
10 interface 708. The physical interface 708 preferably reformats the response into an API message. When a processed packet is forwarded to the physical interface 708 the underlying DSP also interrupts a APITask Thread 710 that retrieves the processed API messages from the physical interface 708. The APITask Thread 710
15 determines whether the API message is an event such as, for example, a voice or fax packet or a DTMF detection which are forwarded directly to the MTA call client 710(a), or a response to a command/request from the MTA call client. For example, the MTA call client may command the voice and data processor to turn
20 off the echo canceller. Such a command is preferably processed by the API interface 706 to add the appropriate header word and forwarded to the physical interface 708. The physical interface 708 formats the message in accordance with the underlying DSP and issues the command the underlying voice channel to turn off the
25 echo canceller. When the command has been complied with a response is returned from the underlying DSP to the physical interface 708. The physical interface 708 formats the response into an API message and forwards it to the APITask thread 710 which then forwards it as an API response to the API interface
30 706. The API interface 706 correlates the API responses received from the APITask thread 710 to the corresponding command/request that prompted the response and forwards a HAPI response to the MTA call client 620.

Referring to FIG. 27, the described channel associated
35 signaling (CAS) logic 632 utilizes a foreign exchange office

1 (FXO) interface to provide exchange of commands and detection of
events between the MTA call client 620 and standard commercial
analog loop/ground start devices such as, for example, plain old
telephone sets (POTS), group three facsimiles, modems, answering
5 machine or other similar devices. The CAS logic 632 can operate
in switch mode, such that ABCD bits 712 are terminated within the
CAS logic. The MTA call client interfaces with the CAS logic
through high level commands 714 and events 716. For example, if
the CAS logic detects a phone going off the hook, a detection
10 event is forwarded to the MTA call client 620 which may then
notify the call management server as previously described. The
MTA call client is also able to issue high level commands to the
CAS logic.

For example, to ring a phone, the MTA call client 620 sends
15 a ring generation profile to the CAS logic, and the command is
switched into raw CAS bits by FXO termination logic 720 to
control the external interface via a hardware abstraction layer
(HAL) 722. Similarly detected events, signaled from the hardware
SLIC 724, including on/off hook, flash hook and potentially pulse
20 dial, are returned to the FXO termination logic 720 from the
hardware abstraction layer as raw CAS bits. The HAL 722 provides
a physical interface to the connected SLIC. The HAL 722 formats
message signals in accordance with the underlying SLIC and
forwards or transmits the message signal to the connected
25 telephony device via the SLIC 724.

The FXO termination logic 720 preferably verifies the
integrity of the detected event and forwards a detected event
message to the MTA call client. The FXO termination logic 720
includes a state machine which monitors detected events signaled
30 from the SLIC 724 via the hardware abstraction layer 722. The FXO
termination logic 720 qualifies detected events signaled by the
SLIC 724 to determine if a valid on/off hook, flash hook or pulse
dial has been detected. The FXO termination logic analyzes the
period of the received signal to determine valid flash hook or
35 pulse dial conditions. Similarly, the FXO termination logic 720

1 includes a state machine which associated with the ringing of
connected phones. When the MTA call client issues a command to
ring a connected phone with a pre-defined cadence and the
associated FXO termination state machine will ring the connected
5 phone in accordance with the MTA command.

The described exemplary voice and data processor is
preferably implemented with a programmable DSP software
architecture (see FIG. 22). The programmable DSP 600 is
effectively hidden within the embedded communications software
10 layer. The software layer binds all core DSP algorithms
together, interfaces the DSP hardware to the host, and provides
low level services such as the allocation of resources to allow
higher level software programs to run. An exemplary multi-layer
software architecture loaded into the program memory 602 for
15 execution on the DSP platform is shown in FIG.28. The MTA call
client 620 provides overall executive control and system
management, and directly interfaces a DSP server 730 to the host
MIPS core (see to FIG. 3). The DSP server 730 provides DSP
resource management and telecommunications signal processing.
20 Operating below the DSP server layer are a number of physical
devices (PXD) 732a, 732b, 732c. Each PXD provides an interface
between the DSP server 730 and an external telephony device (not
shown) via a hardware abstraction layer (HAL) 734.

The DSP server 730 includes a resource manager 736 which
25 receives commands from, forwards events to, and exchanges data
with the MTA call client 620. The user application layer 736
can either be resident on the DSP 600 or alternatively within the
MTA call client. An application programming interface 738 (API)
provides a software interface between the user MTA call client
30 620 and the resource manager 736. The resource manager 736
manages the internal / external program and data memory of the
DSP 600. In addition the resource manager dynamically allocates
DSP resources, performs command routing as well as other general
purpose functions.

1 Recommendation G.711 (1988) - Pulse code modulation (PCM) of
voice frequencies), G.726 16/24/32/40 kbps (ITU Recommendation
G.726 (12/90) - 40, 32, 24, 16 kbit/s Adaptive Differential Pulse
Code Modulation (ADPCM)), G.729A 8 kbps (Annex A (11/96) to ITU
5 Recommendation G.729 - Coding of speech at 8 kbit/s using
conjugate structure algebraic-code-excited linear-prediction (CS-
ACELP) - Annex A: Reduced complexity 8 kbit/s CS-ACELP speech
codec), and G.723 5.3/6.3 kbps (ITU Recommendation G.723.1
(03/96) - Dual rate coder for multimedia communications
10 transmitting at 5.3 and 6.3 kbit/s). The contents of each of the
foregoing ITU Recommendations being incorporated herein by
reference as if set forth in full.

The packet voice exchange 760 is common to both the voice
mode 750 and the voiceband data mode 752. In the voiceband data
15 mode 752, the resource manager invokes the packet voice exchange
760 for exchanging transparently data without modification (other
than packetization) between the telephony device (or circuit
switched network) and the packet based network. This is
typically used for the exchange of fax and modem data when
20 bandwidth concerns are minimal as an alternative to demodulation
and remodulation. During the voiceband data mode 752, the human
speech detector service 768 is also invoked by the resource
manager. The human speech detector 768 monitors the signal from
the near end telephony device for speech. In the event that
25 speech is detected by the human speech detector 768, an event is
forwarded to the resource manager which, in turn, causes the
resource manager to terminate the human speech detector service
768 and invoke the appropriate services for the voice mode 750
(i.e., the call discriminator, the packet tone exchange, and the
30 packet voice exchange).

In the fax relay mode 754, the resource manager invokes a
packet fax data exchange 764 service. The packet fax exchange 764
may employ various data pumps including, among others, V.17 which
can operate up to 14,400 bits per second, V.29 which uses a 1700-
35 Hz carrier that is varied in both phase and amplitude, resulting

1 in 16 combinations of 8 phases and 4 amplitudes which can operate
up to 9600 bits per second, and V.27ter which can operate up to
4800 bits per second. Likewise, the resource manager invokes a
packet modem data exchange 766 service in the data relay mode
5 756. The packet modem data exchange 766 may employ various data
pumps including, among others, V.22bis/V.22 with data rates up
to 2400 bits per second, V.32bis/V.32 which enables full-duplex
transmission at 14,400 bits per second, and V.34 which operates
up to 33,600 bits per second. The ITU Recommendations setting
10 forth the standards for the foregoing data pumps are incorporated
herein by reference as if set forth in full.

In the described exemplary embodiment, the user application
layer does not need to manage any service directly. The user
application layer manages the session using high-level commands
15 directed to the NetVHD, which in turn directly runs the services.
However, the user application layer can access more detailed
parameters of any service if necessary to change, by way of
example, default functions for any particular application.

In operation, the user application layer opens the NetVHD
20 and connects it to the appropriate PXD. The user application
then may configure various operational parameters of the NetVHD,
including, among others, default voice compression (Linear,
G.711, G.726, G.723.1, G.723.1A, G.729A, G.729B), fax data pump
(Binary, V.17, V.29, V.27ter), and modem data pump (Binary,
25 V.22bis, V.32bis, V.34). The user application layer then loads
an appropriate signaling service (not shown) into the NetVHD,
configures it and sets the NetVHD to the On-hook state.

In response to events from the signaling service (not shown)
via a near end telephony device (hookswitch), or signal packets
30 from the far end, the user application will set the NetVHD to
the appropriate off-hook state, typically voice mode. In an
exemplary embodiment, if the signaling service event is triggered
by the near end telephony device, the packet tone exchange will
generate dial tone. Once a DTMF tone is detected, the dial tone
35 is terminated. The DTMF tones are packetized and forwarded to

1 voice to be carried over traditional media such as time division
multiplex (TDM) networks and voice storage and playback systems.

The PXDs for the voice mode provide echo cancellation, gain,
and automatic gain control. The network VHD invokes numerous
5 services in the voice mode including call discrimination, packet
voice exchange, and packet tone exchange. These network VHD
services operate together to provide: (1) an encoder system with
DTMF detection, call progress tone detection, voice activity
detection, voice compression, and comfort noise estimation, and
10 (2) a decoder system with delay compensation, voice decoding,
DTMF generation, comfort noise generation and lost frame
recovery.

The services invoked by the network VHD in the voice mode
and the associated PXD is shown schematically in FIG. 30. In the
15 described exemplary embodiment, the PXD 1060 provides two way
communication with a telephone or a circuit switched network,
such as a PSTN line (e.g. DS0) carrying a 64kb/s pulse code
modulated (PCM) signal, i.e., digital voice samples.

The incoming PCM signal 1060a is initially processed by the
20 PXD 1060 to remove far end echos. As the name implies, echos in
telephone systems is the return of the talker's voice resulting
from the operation of the hybrid with its two-four wire
conversion. If there is low end-to-end delay, echo from the far
end is equivalent to side-tone (echo from the near-end), and
25 therefore, not a problem. Side-tone gives users feedback as to
how loud they are talking, and indeed, without side-tone, users
tend to talk too loud. However, far end echo delays of more than
about 10 to 30 msec significantly degrade the voice quality and
are a major annoyance to the user.

30 An echo canceller 1070 is used to remove echos from far end
speech present on the incoming PCM signal 1060a before routing
the incoming PCM signal 1060a back to the far end user. The echo
canceller 1070 samples an outgoing PCM signal 1060b from the far
end user, filters it, and combines it with the incoming PCM
35 signal 1060a. Preferably, the echo canceller 1070 is followed

during these periods. A VAD 1080, operating under the packet voice exchange, is used to accomplish this function. The VAD 1080 attempts to detect digital voice samples that do not contain active speech. During periods of inactive speech, the comfort noise estimator 1081 couples silence identifier (SID) packets to a packetization engine 1078. The SID packets contain voice parameters that allow the reconstruction of the background noise at the far end.

From a system point of view, the VAD 1080 may be sensitive to the change in the NLP 1072. For example, when the NLP 1072 is activated, the VAD 1080 may immediately declare that voice is inactive. In that instance, the VAD 1080 may have problems tracking the true background noise level. If the echo canceller 1070 generates comfort noise during periods of inactive speech, it may have a different spectral characteristic from the true background noise. The VAD 1080 may detect a change in noise character when the NLP 1072 is activated (or deactivated) and declare the comfort noise as active speech. For these reasons, the VAD 1080 should be disabled when the NLP 1072 is activated. This is accomplished by a "NLP on" message 1072a passed from the NLP 1072 to the VAD 1080.

The voice encoder 1082, operating under the packet voice exchange, can be a straight 16 bit PCM encoder or any voice encoder which supports one or more of the standards promulgated by ITU. The encoded digital voice samples are formatted into a voice packet (or packets) by the packetization engine 1078. These voice packets are formatted according to an applications protocol and outputted to the host (not shown). The voice encoder 1082 is invoked only when digital voice samples with speech are detected by the VAD 1080. Since the packetization interval may be a multiple of an encoding interval, both the VAD 1080 and the packetization engine 1078 should cooperate to decide whether or not the voice encoder 1082 is invoked. For example, if the packetization interval is 10 msec and the encoder interval is 5 msec (a frame of digital voice samples is 5 ms), then a

1 Similarly, a call progress tone detector 1077 also operates
under the packet tone exchange to determine whether a precise
signaling tone is present at the near end. Call progress tones
are those which indicate what is happening to dialed phone calls.
5 Conditions like busy line, ringing called party, bad number, and
others each have distinctive tone frequencies and cadences
assigned them. The call progress tone detector 1077 monitors the
call progress state, and forwards a call progress tone signal to
the packetization engine to be packetized and transmitted across
10 the packet based network. The call progress tone detector may
also provide information regarding the near end hook status which
is relevant to the signal processing tasks. If the hook status
is on hook, the VAD should preferably mark all frames as
inactive, DTMF detection should be disabled, and SID packets
15 should only be transferred if they are required to keep the
connection alive.

The decoding system of the network VHD 1062 essentially
performs the inverse operation of the encoding system. The
decoding system of the network VHD 1062 comprises a depacketizing
20 engine 1084, a voice queue 1086, a DTMF queue 1088, a precision
tone queue 1087, a voice synchronizer 1090, a DTMF synchronizer
1102, a precision tone synchronizer 1103, a voice decoder 1096,
a VAD 1098, a comfort noise estimator 1100, a comfort noise
generator 1092, a lost packet recovery engine 1094, a tone
25 generator 1104, and a precision tone generator 1105.

The depacketizing engine 1084 identifies the type of packets
received from the host (i.e., voice packet, DTMF packet, call
progress tone packet, SID packet), transforms them into frames
which are protocol independent. The depacketizing engine 1084
30 then transfers the voice frames (or voice parameters in the case
of SID packets) into the voice queue 1086, transfers the DTMF
frames into the DTMF queue 1088 and transfers the call progress
tones into the call progress tone queue 1087. In this manner,
the remaining tasks are, by and large, protocol independent.

1 A jitter buffer is utilized to compensate for network
impairments such as delay jitter caused by packets not arriving
at the same time or in the same order in which they were
transmitted. In addition, the jitter buffer compensates for lost
5 packets that occur on occasion when the network is heavily
congested. In the described exemplary embodiment, the jitter
buffer for voice includes a voice synchronizer 1090 that operates
in conjunction with a voice queue 1086 to provide an isochronous
stream of voice frames to the voice decoder 1096.

10 Sequence numbers embedded into the voice packets at the far
end can be used to detect lost packets, packets arriving out of
order, and short silence periods. The voice synchronizer 1090
can analyze the sequence numbers, enabling the comfort noise
generator 1092 during short silence periods and performing voice
15 frame repeats via the lost packet recovery engine 1094 when voice
packets are lost. SID packets can also be used as an indicator
of silent periods causing the voice synchronizer 1090 to enable
the comfort noise generator 1092. Otherwise, during far end
active speech, the voice synchronizer 1090 couples voice frames
20 from the voice queue 1086 in an isochronous stream to the voice
decoder 1096. The voice decoder 1096 decodes the voice frames
into digital voice samples suitable for transmission on a circuit
switched network, such as a 64kb/s PCM signal for a PSTN line.
The output of the voice decoder 1096 (or the comfort noise
25 generator 1092 or lost packet recovery engine 1094 if enabled)
is written into a media queue 1106 for transmission to the PXD
1060.

The comfort noise generator 1092 provides background noise
to the near end user during silent periods. If the protocol
30 supports SID packets, (and these are supported for VTOA, FRF-11,
and VoIP), the comfort noise estimator at the far end encoding
system should transmit SID packets. Then, the background noise
can be reconstructed by the near end comfort noise generator 1092
from the voice parameters in the SID packets buffered in the
35 voice queue 1086. However, for some protocols, namely, FRF-11,

1 digit gap before call progress tones are generated. This is achieved by a "tone present" message 1087a passed between the call progress tone queue 1087 and the voice synchronizer 1090.

5 The call progress tone generator 1105 converts the call progress tone signals into a call progress tone suitable for a standard digital or analog telephone. The call progress tone generator 1105 overwrites the media queue 1106 to prevent leakage through the voice path and to ensure that the call progress tones are not too noisy.

10 The outgoing PCM signal in the media queue 1106 is coupled to the PXD 1060 via the switchboard 1032'. The outgoing PCM signal is coupled to an amplifier 1108 before being outputted on the PCM output line 1060b.

15 An exemplary voice signal processor is disclosed U.S. Patent Application No. 09/522,185, entitled "Voice and Data Exchange Over a Packet Based Network," the contents of which is hereby incorporated by reference as though fully set forth herein.

B. The Fax Relay Mode

20 Fax relay mode provides signal processing of fax signals. Fax relay mode enables the transmission of fax signals over a packet based system such as VoIP, VoFR, FRF-11, VTOA, or any other proprietary network. For the purposes of explanation, first fax machine is called a sending fax that is connected to the sending network gateway 1378a through a PSTN. The sending network gateway is connected to a CMTS via a HFC network. 25 Additional fax machines may be on line connections coupled to the other end of the CMTS via a network gateway and a HFC network, or off line connections, coupled to the CMTS for example by a telephone network gateway and a PSTN.

30 The transfer of fax signals over packet based networks may be accomplished by at least three alternative methods. In the first method, fax data signals are exchanged in real time. Typically, the sending and receiving fax machines are spoofed to allow transmission delays plus jitter of up to about 1.2 seconds.

35 The second, store and forward mode, is a non real time method of

09:34:30

1 transferring fax data signals. Typically, the fax communication
is transacted locally, stored into memory and transmitted to the
destination fax machine at a subsequent time. The third mode is
a combination of store and forward mode with minimal spoofing to
5 provide an approximate emulation of a typical fax connection.

In the fax relay mode, the network VHD invokes the packet
fax data exchange. The packet fax data exchange provides
demodulation and re-modulation of fax data signals. This
approach results in considerable bandwidth savings since only the
10 underlying unmodulated data signals are transmitted across the
packet based network. The packet fax data exchange also provides
compensation for network jitter with a jitter buffer similar to
that invoked in the packet voice exchange. Additionally, the
packet fax data exchange compensates for lost data packets with
15 error correction processing. Spoofing may also be provided
during various stages of the procedure between the fax machines
to keep the connection alive.

The packet fax data exchange is divided into two basic
functional units, a demodulation system and a re-modulation
20 system. In the demodulation system, the network VHD couples fax
data signals from a circuit switched network, or a fax machine,
to the packet based network. In the re-modulation system, the
network VHD couples fax data signals from the packet network to
the switched circuit network, or a fax machine directly.

25 During real time relay of fax data signals over a packet
based network, the sending and receiving fax machines are spoofed
to accommodate network delays plus jitter. Typically, the packet
fax data exchange can accommodate a total delay of up to about
1.2 seconds. Preferably, the packet fax data exchange supports
30 error correction mode (ECM) relay functionality, although a full
ECM implementation is typically not required. In addition, the
packet fax data exchange should preferably preserve the typical
call duration required for a fax session over a PSTN/ISDN when
exchanging fax data signals between two terminals.

35

1 The packet fax data exchange for the real time exchange of
fax data signals between a circuit switched network and a packet
based network is shown schematically in FIG. 46. In this
exemplary embodiment, a connecting PXD (not shown) connecting the
5 fax machine to the switch board 1032' is transparent, although
those skilled in the art will appreciate that various signal
conditioning algorithms could be programmed into PXD such as echo
cancellation and gain.

After the PXD (not shown), the incoming fax data signal
10 1390a is coupled to the demodulation system of the packet fax
data exchange operating in the network VHD via the switchboard
1032'. The incoming fax data signal 1390a is received and
buffered in an ingress media queue 1390. A V.21 data pump 1392
demodulates incoming T.30 message so that T.30 relay logic 1394
15 can decode the received T.30 messages 1394a. Local T.30
indications 1394b are packetized by a packetization engine 1396
and if required, translated into T.38 packets via a T.38 shim
1398 for transmission to a T.38 compliant remote network gateway
(not shown) across the packet based network. The V.21 data pump
20 1392 is selectively enabled/disabled 1394c by the T.30 relay
logic 1394 in accordance with the reception/ transmission of the
T.30 messages or fax data signals. The V.21 data pump 1392 is
common to the demodulation and re-modulation system. The V.21
data pump 1392 communicates T.30 messages such as for example
25 called station tone (CED) and calling station tone (CNG) to
support fax setup between a local fax device (not shown) and a
remote fax device (not shown) via the remote network gateway.

The demodulation system further includes a receive fax data
pump 1400 which demodulates the fax data signals during the data
30 transfer phase. The receive fax data pump 1400 supports the
V.27ter standard for fax data signal transfer at 2400/4800 bps,
the V.29 standard for fax data signal transfer at 7200/9600 bps,
as well as the V.17 standard for fax data signal transfer at
7200/9600/12000/14400 bps. The V.34 fax standard, once approved,
35 may also be supported. The T.30 relay logic 1394 enables /

1 In data relay mode, a local modem connection is established
on each end of the packet based network 1494. That is, the call
modem 1492a and the call network gateway 496a establish a local
modem connection, as does the destination answer modem 1492b and
5 its respective answer network gateway 1496b. Next, data signals
are relayed across the packet based network 1494. The call
network gateway 1496a demodulates the data signal and formats the
demodulated data signal for the particular packet based network
1494. The answer network gateway 1496b compensates for network
10 impairments and remodulates the encoded data in a format suitable
for the destination answer modem 1492b. This approach results
in considerable bandwidth savings since only the underlying
demodulated data signals are transmitted across the packet based
network.

15 In the data relay mode, the packet data modem exchange
provides demodulation and modulation of data signals. With full
duplex capability, both modulation and demodulation of data
signals can be performed simultaneously. The packet data modem
exchange also provides compensation for network jitter with a
20 jitter buffer similar to that invoked in the packet voice
exchange. Additionally, the packet data modem exchange
compensates for system clock jitter between modems with a dynamic
phase adjustment and resampling mechanism. Spoofing may also be
provided during various stages of the call negotiation procedure
25 between the modems to keep the connection alive.

The packet data modem exchange invoked by the network VHD
in the data relay mode is shown schematically in FIG. 49. In the
described exemplary embodiment, a connecting PXD (not shown)
connecting a modem to the switch board 1032' is transparent,
30 although those skilled in the art will appreciate that various
signal conditioning algorithms could be programmed into the PXD
such as filtering, echo cancellation and gain.

After the PXD, the data signals are coupled to the network
VHD via the switchboard 1032'. The packet data modem exchange
35 provides two way communication between a circuit switched network

1 and packet based network with two basic functional units, a
demodulation system and a remodulation system. In the
demodulation system, the network VHD exchanges data signals from
a circuit switched network, or a telephony device directly, to
5 a packet based network. In the remodulation system, the network
VHD exchanges data signals from the packet based network to the
PSTN line, or the telephony device.

In the demodulation system, the data signals are received
and buffered in an ingress media queue 1500. A data pump
10 receiver 1504 demodulates the data signals from the ingress media
queue 1500. The data pump receiver 1504 supports the V.22bis
standard for the demodulation of data signals at 1200/2400 bps;
the V.32bis standard for the demodulation of data signals at
4800/7200/9600/12000/14400 bps, as well as the V.34 standard for
15 the demodulation of data signals up to 33600 bps. Moreover, the
V.90 standard may also be supported. The demodulated data
signals are then packetized by the packetization engine 1506 and
transmitted across the packet based network.

In the remodulation system, packets of data signals from the
20 packet based network are first depacketized by a depacketizing
engine 1508 and stored in a jitter buffer 1510. A data pump
transmitter 1512 modulates the buffered data signals with a
voiceband carrier. The modulated data signals are in turn stored
in the egress media queue 1514 before being output to the PXD
25 (not shown) via the switchboard 1032'. The data pump transmitter
1512 supports the V.22bis standard for the transfer of data
signals at 1200/2400 bps; the V.32bis standard for the transfer
of data signals at 4800/7200/9600/12000/14400 bps, as well as the
V.34 standard for the transfer of data signal up to 33600 bps.
30 Moreover, the V.90 standard may also be supported.

During jitter buffer underflow, the jitter buffer 1510 sends
a buffer low indication 1510a to spoofing logic 1516. When the
spoofing logic 1516 receives the buffer low signal indicating
that the jitter buffer 1510 is operating below a predetermined
35 threshold level, it inserts spoofed data at the appropriate place

in the data signal via the data pump transmitter 1512. Spoofing continues until the jitter buffer 1510 is filled to the predetermined threshold level, at which time data signals are again transferred from the jitter buffer 1510 to the data pump transmitter 1512.

End to end clock logic 1518 also monitors the state of the jitter buffer 1510. The clock logic 1518 controls the data transmission rate of the data pump transmitter 1512 in correspondence to the state of the jitter buffer 1510. When the jitter buffer 1510 is below a predetermined threshold level, the clock logic 1518 reduces the transmission rate of the data pump transmitter 1512. Likewise, when the jitter buffer 1510 is above a predetermined threshold level, the clock logic 1518 increases the transmission rate of the data pump transmitter 1512.

Before the transmission of data signals across the packet based network, the connection between the two modems must first be negotiated through a handshaking sequence. This entails a two-step process. First, a call negotiator 1502 determines the type of modem (i.e., V.22, V.32bis, V.34, V.90, etc.) connected to each end of the packet based network. Second, a rate negotiator 1520 negotiates the data signal transmission rate between the two modems.

The call negotiator 1502 determines the type of modem connected locally, as well as the type of modem connected remotely via the packet based network. The call negotiator 1502 utilizes V.25 automatic answering procedures and V.8 auto-baud software to automatically detect modem capability. The call negotiator 1502 receives protocol indication signals 1502a (ANSam and V.8 menus) from the ingress media queue 1500, as well as AA, AC and other message indications 1502b from the local modem via a data pump state machine 1522, to determine the type of modem in use locally. The call negotiator 1502 relays the ANSam answer tones and other indications 1502e from the data pump state machine 1522 to the remote modem via a packetization engine 1506. The call negotiator also receives ANSam, AA, AC and other

indications 1502c from a remote modem (not shown) located on the opposite end of the packet based network via a depacketizing engine 1508. The call negotiator 1502 relays ANSam answer tones and other indications 1502d to a local modem (not shown) via an egress media queue 1514 of the modulation system. With the ANSam, AA, AC and other indications from the local and remote modems, the call negotiator 1502 can then negotiate a common standard (i.e., V.22, V.32bis, V.34, V.90, etc.) in which the data pumps must communicate with the local modem and the remote modems.

The packet data modem exchange preferably utilizes indication packets as a means for communicating answer tones, AA, AC and other indication signals across the packet based network. However, the packet data modem exchange supports data pumps such as V.22bis and V.32bis which do not include a well defined error recovery mechanism, so that the modem connection may be terminated whenever indication packets are lost. Therefore, either the packet data modem exchange or the application layer should ensure proper delivery of indication packets when operating in a network environment that does not guarantee packet delivery.

The packet data modem exchange can ensure delivery of the indication packets by periodically retransmitting the indication packet until some expected packets are received. For example, in V.32bis relay, the call negotiator operating under the packet data modem exchange on the answer network gateway periodically retransmits ANSam answer tones from the answer modem to the call modem, until the calling modem connects to the line and transmits carrier state AA.

Alternatively, the packetization engine can embed the indication information directly into the packet header. In this approach, an alternate packet format is utilized to include the indication information. During modem handshaking, indication packets transmitted across the packet based network include the indication information, so that the system does not rely on the

1 successful transmission of individual indication packets.
Rather, if a given packet is lost, the next arriving packet
contains the indication information in the packet header. Both
methods increase the traffic across the network. However, it is
5 preferable to periodically retransmit the indication packets
because it has less of a detrimental impact on network traffic.

A rate negotiator 1520 synchronizes the connection rates at
the network gateways 1496a, 1496b, 1496c (see FIG. 49). The rate
negotiator receives rate control codes 1520a from the local modem
10 via the data pump state machine 1522 and rate control codes 1520b
from the remote modem via the depacketizing engine 1508. The
rate negotiator 1520 also forwards the remote rate control codes
1520a received from the remote modem to the local modem via
commands sent to the data pump state machine 1522. The rate
15 negotiator 1520 forwards the local rate control codes 520c
received from the local modem to the remote modem via the
packetization engine 1506. Based on the exchanged rate codes the
rate negotiator 1520 establishes a common data rate between the
calling and answering modems. During the data rate exchange
20 procedure, the jitter buffer 1510 should be disabled by the rate
negotiator 1520 to prevent data transmission between the call
and answer modems until the data rates are successfully
negotiated.

Similarly error control (V.42) and data compression
25 (V.42bis) modes should be synchronized at each end of the packet
based network. Error control logic 1524 receives local error
control messages 1524a from the data pump receiver 1504 and
forwards those V.14/V.42 negotiation messages 1524c to the remote
modem via the packetization engine 1506. In addition, error
30 control logic 1524 receives remote V.14/V.42 indications 1524b
from the depacketizing engine 1508 and forwards those V.14/V.42
indications 1524d to the local modem. With the V.14/V.42
indications from the local and remote modems, the error control
logic 1524 can negotiate a common standard to ensure that the
35 network gateways utilize a common error protocol. In addition,

1 error control logic 1524, communicates the negotiated error control protocol 1524(e) to the spoofing logic 1516 to ensure data mode spoofing is in accordance with the negotiated error control mode.

5 V.42 is a standard error correction technique using advanced cyclical redundancy checks and the principle of automatic repeat requests (ARQ). In accordance with the V.42 standard, transmitted data signals are grouped into blocks and cyclical redundancy calculations add error checking words to the transmitted data signal stream. The receiving modem calculates new error check information for the data signal block and compares the calculated information to the received error check information. If the codes match, the received data signals are valid and another transfer takes place. If the codes do not match, a transmission error has occurred and the receiving modem requests a repeat of the last data block. This repeat cycle continues until the entire data block has been received without error.

15 Various voiceband data modem standards exist for error correction and data compression. V.42bis and MNP5 are examples of data compression standards. The handshaking sequence for every modem standard is different so that the packet data modem exchange should support numerous data transmission standards as well as numerous error correction and data compression techniques.

20 An exemplary data relay is disclosed U.S. Patent Application No. 09/522,185, entitled "Voice and Data Exchange Over a Packet Based Network," the contents of which has been previously incorporated herein by reference.

30 Although a preferred embodiment of the present invention has been described, it should not be construed to limit the scope of the appended claims. For example, the present invention can be implemented by both a software embodiment or a hardware embodiment. Those skilled in the art will understand that various modifications may be made to the described embodiment.

35

1 Moreover, to those skilled in the various arts, the invention
itself herein will suggest solutions to other tasks and
adaptations for other applications. It is therefore desired that
the present embodiments be considered in all respects as
5 illustrative and not restrictive, reference being made to the
appended claims rather than the foregoing description to indicate
the scope of the invention.

10

15

20

25

30

35